CLAIMS

1. (Currently amended) A system for processing a set of data bits comprising:

storage means for storing ~~the set of data bits~~ a current state of a subset of a recurring serial sequence of scramble bits;

digital logic means for determining ~~a~~ the current state of the subset of ~~a~~ the serial sequence of scramble bits by applying a generating polynomial to the serial sequence of scramble bits;

generating means for generating ~~the~~ a next state of the subset ~~responsive to an immediately preceding state of the subset~~ by logically manipulating at least one bit of the current state of the subset with at least another bit of the current state of the subset; and

digital operation means for performing a bitwise parallel digital operation between each bit of the set of data bits with at least one corresponding bit of the current state of the subset to produce an output set of data bits;

~~where a number of bits in the subset corresponds to a periodicity of the serial sequence of scramble bits~~

where the storage means is configured to replace the stored current state of the subset with the next state of the subset after the digital operation means performs the bitwise parallel digital operation.

2. (Currently amended) ~~A~~ The system according to claim 1 where the system is adapted to scramble the set of data bits using the current state of the subset.

3. (Currently amended) ~~A~~ The system according to claim 1 where the system is adapted to descramble the set of data bits using the current state of the subset.

4. (Currently amended) ~~A~~ The system according to claim 1 comprising receiving means including multiplexing means for receiving the serial sequence of scramble bits.

5. (Canceled)

6. (Currently amended) ~~A~~ The system according to claim 1 where the digital logic means includes combinational logic means for logically manipulating the serial sequence of scramble bits.

7. (Currently amended) ~~A~~ The system according to claim 1 where the bitwise parallel operation includes a bitwise parallel XOR operation.

8. (Currently amended) A digital scrambler/descrambler using a subset of a serial sequence of scrambler bits, the scrambler/descrambler comprising:

selection means for selecting between a first set of data bits to be scrambled and a second set of data bits to be descrambled;

digital logic means for determining a subset of the serial sequence of scrambler bits, the subset being determined ~~based on an immediately preceding subset of the serial sequence of scrambler bits~~ by logically manipulating at least one bit of a preceding state of the subset with at least another bit of the preceding state of the subset;

digital operation means for executing a bitwise parallel digital operation between the subset and either the first or the second set of data bits;

where a number of bits in the subset corresponds to a periodicity of the serial sequence of scramble bits.

9. (Currently amended) ~~A~~ The digital scrambler/descrambler according to claim 8 where the bitwise parallel digital operation includes a bitwise parallel XOR operation.

10. (Currently amended) ~~A~~ The digital scrambler/descrambler according to claim 8 where the selection means includes a multiplexer.

11. (Currently amended) ~~A~~ The digital scrambler/descrambler according to claim 8 where the digital logic means includes a combinational logic circuit.

12. (Currently amended) ~~A~~ The digital scrambler/descrambler according to claim 11 where- the digital logic means includes a digital storage means for storing the immediately

preceding subset.

13. (Currently amended) A method of processing a plurality of data bits using a subset of a recurring serial sequence of scrambler bits, the method comprising:

a) storing in parallel the plurality of data bits;

b) determining a subset of the recurring serial sequence of scrambler bits ~~based on an immediately preceding subset of the recurring serial sequence of scrambler bits~~ by logically manipulating at least one bit of a preceding state of the subset with at least another bit of the preceding state of the subset;

c) generating the subset where, for each bit of the plurality of data bits, at least one bit of the appropriate subset is associated therewith; and

d) performing a bitwise parallel XOR operation between each bit of the plurality of data bits and the at least one bit of the subset associated therewith to produce an output set of data bits;

where a number of bits in the subset corresponds to a periodicity of the recurring serial sequence of scramble bits.

14. (Currently amended) ~~A~~ The method according to claim 13 comprising performing logical operations between specific scrambler bits of the immediately preceding subset.

15. (Canceled)

16. (Currently amended) ~~A~~ The method according to claim 13 comprising loading the subset in a register.

17. (Currently amended) A scrambler comprising:

a register block to store a current state of a subset of a recurring serial sequence of scramble bits;

a predict logic block to generate the current state of the subset ~~responsive to an immediately preceding state of the subset~~ by logically manipulating at least one bit of an immediately

preceding state of the subset with at least another bit of the immediately preceding state of the subset; and

a scramble logic block to scramble a data set in parallel with the current state of the subset;

where the number of bits in the current state of the subset corresponds to a periodicity of the recurring serial sequence of scramble bits.

18. (Previously presented) The scrambler of claim 17 comprising a multiplexer block to multiplex a serial version of the data set to create a parallel version of the data set.

19. (Previously presented) The scrambler of claim 17 where the predict logic block or the scramble logic block includes a corresponding plurality of parallel combinational sub blocks.

20. (Currently amended) The scrambler of claim 17 where the predict logic block is configured to determine the current state of the subset by applying a generator polynomial ~~defined in the IEEE 802.11a~~.